CLAIMS

Claims 1-4: (cancelled).

5. (currently amended) A system for use in developing risk management software, said system comprising an object-oriented library of library objects stored on a computer-readable storage medium, said library comprising:

(a) a first plurality of library objects for use in generating scenario~~s~~ data, comprising a plurality of generators passable to at least one software component that requires data provided by said generators, wherein each of said plurality of generators produces elements of a sequence governed by a sequential process, wherein each generator comprises a common generator interface, where said common generator interface comprises a first function that generates the next element of said sequence and a second function that returns the element most recently generated by said first function;

(b) a second plurality of library objects for calculating instrument values, comprising at least one of

i. a plurality of maps, wherein each map comprises a common map interface, wherein said common map interface comprises a third function that receives as input a first object of a first type, transforms said first object according to a pre-defined rule to produce a second object of a second type, and returns said second object, and

ii. a plurality of accumulators, wherein each accumulator comprises an internal state and a common accumulator interface, said internal state having a current value, said current value being a function of one or more past values of said internal state and one or more data values received as input by said accumulator, said common accumulator interface comprising a fourth function to receive as input said one or more data values and to update said internal state to a new current value, and a fifth function that returns a value corresponding to the current value of the internal state; and

(c) a third plurality of library objects comprising a plurality of operations, said plurality of operations defined to operate on at least one generator of said plurality of generators and at least one other operand in an expression to produce a new generator;

said new generator for use in performing a simulation to obtain prices for a plurality of instruments over a plurality of scenarios.

6. (original) The system as claimed in claim 5, wherein each generator is adapted to produce elements of either a deterministic or stochastic sequence.

7. (original) The system as claimed in claim 5, wherein said other operand is selected from the following group: a generator; a map; an accumulator; and objects of compatible type such that said arithmetic expression is defined.

8. (original) The system as claimed in claim 5, wherein said expression is an arithmetic expression comprising one or more of the arithmetic operations selected from the following group: addition, subtraction, multiplication, and division.

9. (original) The system as claimed in claim 5, wherein said expression comprises a composition operation.

10. (currently amended) A method of developing risk management software using objects from an object-oriented library, said software comprising instructions for simulating a portfolio of financial instruments over a plurality of scenarios, said method comprising the steps of:

(a) constructing a scenario generator passable to at least one software component that requires data provided by said scenario generator, wherein said scenario generator produces elements of a sequence governed by a sequential process, wherein said scenario generator comprises a common generator interface, and wherein said common generator interface comprises a first function that generates the next element of said sequence and a second function that returns the element most recently generated by said first function;

(b) constructing one or more pricing maps, wherein each pricing map is associated with an instrument in a portfolio, wherein each pricing map comprises a common map interface, wherein said common map interface comprises a third function that receives as input a first object of a first type, transforms said first object according to a pre-defined rule to produce a second object of a second type, and returns said object;

(c) composing said one or more pricing maps with said scenario generator to obtain a pricing generator; and

(d) using said pricing generator in performing a simulation to obtain prices for a plurality of instruments.

11. (original) The method as claimed in claim 10, wherein step (c) comprises the steps of composing said one or more pricing maps to produce a composite pricing map, and composing said composite pricing map with said scenario generator to obtain a pricing generator.

12. (original) The method as claimed in claim 10, wherein said prices for said plurality of instruments are used to populate a Mark-to-Future cube.

13. (currently amended) A method of developing risk management software using objects from an object-oriented library, said software comprising instructions for simulating a portfolio of financial instruments over a plurality of scenarios, said method comprising the steps of:

(a) constructing a stochastic sequence generator passable to at least one software component that requires data provided by said scenario generator, wherein said stochastic sequence generator produces elements of a sequence governed by a sequential process, wherein said stochastic sequence generator comprises a common generator interface, where said common generator interface comprises a first function that generates the next element of said sequence and a second function that returns the element most recently generated by said first function;

(b) constructing one or more pricing accumulators, wherein each pricing accumulator is associated with an instrument in a portfolio, wherein each pricing accumulator comprises an internal state and a common accumulator interface, said internal state having a current value, said current value being a function of one or more past values of said internal state and one or more data values received as input by said accumulator, said common accumulator interface comprising a third function to receive as input said one or more data values and to update said internal state to a new current value, and a fourth function that returns a value corresponding to the current value of the internal state;

(c) composing said one or more pricing accumulators with said stochastic sequence generator to obtain a pricing generator;

(d) using said pricing generator in performing a simulation to obtain prices for a plurality of instruments at each time step in a scenario; and

(e) repeating step (d) over a plurality of scenarios.

14. (original) The method as claimed in claim 13, wherein step (c) comprises the steps of composing said one or more pricing accumulators to produce a composite pricing accumulator, and composing said composite pricing accumulator with said stochastic sequence generator to obtain a pricing generator.

15. (original) The method as claimed in claim 13, wherein said prices for said plurality of instruments are used to populate a Mark-to-Future cube.

16. (original) The method as claimed in claim 13, wherein step (e) is performed by cloning said pricing generator.